Eliminating Design Errors using Simulink Design Verifier

Nishaat Vasi
Simulink Design Verifier Product Manager
I ran into two unexpected issues with the new Cruise Control feature:

1. I was trying to see how quickly the speed would increase when I modified the holdrate calibration. Originally it was (4) and it worked pretty well. I increased it to (10), within the calibration limits resulted in a runaway condition. After setting the speed, when I increased the speed setting it just kept increasing beyond the 90 mph limit. I finally hit the brake once it hit 100 mph. Good thing I was on the track!

2. I reset the hold rate back to (4) and set the target speed to 40 mph. I was going downhill on the track so I reduced the target speed setting to the limit of 20 mph. Then next time I hit the reduce speed button it set the target speed to 33 mph one time and then another time it was set to 29 mph.

I doubt if either of these behaviors are in the requirements 😊
Gaining Confidence in Your Design

Is this enough?

- Simulate
- Test functional correctness
- Test nominal calibration
- Measure model coverage

Confidence vs. Effort / Time
Questions That Haunt

- Are we missing requirements?
- Do the test vectors adequately cover requirements?
- How about unintended interaction of calibration parameters?
1. **Architecture**

1.1.1. **Enable/Disable Switch**

   The controller shall have an On/Off Switch to enable/disable the operation of the cruise control system.

1.1.2. **Set Speed/Decelerate Button**

   The controller shall have an input button to:
   - set the target speed to the current speed when the cruise control is **not engaged** (active)
   - decelerate (reduce) the target speed when the cruise control is **engaged** (active)

1.1.3. **Resume Speed/Accelerate Button**

   The controller shall have an input button to:
   - set the target speed to last acceptable target speed when the cruise control is **not engaged** (active)
   - accelerate (increase) the target speed when the cruise control is **active**

2. **Functional Requirements**

2.1.1. **Disabled (off) during start-up**

   Initial state of cruise control system shall be disabled.

2.1.2. **Not engaged with enabling (on)**

   The cruise control system shall not be initially engaged with enabling.

2.1.3. **Disengaged (not active) when disabled (off)**
Enhancing Robustness of Design

Parameter “holdrate” can take on a zero value!
Finding Unintended Behavior Early

Detect hard-to-find design errors before simulation

- Dead logic
- Division by zero
- Range violation
- Integer overflow
- Assertion violation
- Out of bound array access
Adding Confidence to Design

- Simulate
- Test functional correctness
- Test nominal calibration
- Measure model coverage
- Check for divide by zero, overflows

Confidence vs. Effort / Time
Field Calibration Tests Uncover Error

**Problem:** “holdrate” cal = 10 sets off runaway acceleration, max cruise speed is exceeded
- Functional tests pass for model
- No redundancies in model (100% coverage achieved)
- Nominal signal and parameter values worked in simulation

**Debug Options:**
1. Create test to reach this Cal condition
2. Ask Simulink Design Verifier if such unintended acceleration can occur
Identify Failure Mode in Model

- Construct a “property” for Simulink Design Verifier
- Leverage Calibration parameter values from field test
- Ask tool to prove whether errant condition can occur
- Use generated test case to debug
Other Issues That Could Be Identified

- Going downhill, minimum coast speed not honored
Adding Confidence to Design

- Simulate
- Test functional correctness
- Test nominal calibration
- Measure model coverage
- Intended behavior checks
- Unintended behavior, robustness checks

Effort / Time

Confidence

Simulink Design Verifier
What is Simulink Design Verifier?

- Design analysis tool
- Identifies robustness errors early
- Proves model meets key requirements under all circumstances
Questions

Please visit our booth for additional demonstration